modulo\_03.R

renato

2021-11-17

# Insumo-Producto Ambientalmente Extendido  
# Facilitado por Renato Vargas  
# Modulo 03 - Matriz de Insumo Producto y   
# Cuentas Ambientales de Costa Rica  
# Publicadas por el Banco Central de Costa Rica  
# Año 2017  
  
# Preámbulo  
library(openxlsx)

## Warning: package 'openxlsx' was built under R version 4.0.5

library(reshape2)  
library(Matrix.utils)

## Warning: package 'Matrix.utils' was built under R version 4.0.5

## Loading required package: Matrix

rm( list = ls() )  
  
  
# Directorio de trabajo (ruta a los datos con "/" en vez de "\")  
wd <- c("D:/github/insumo-producto-ambiental/datos")  
setwd(wd)  
getwd()

## [1] "D:/github/insumo-producto-ambiental/datos"

# Consumo intermedio  
Z\_cruda <- as.matrix(read.xlsx("MIP-AE-AE-017-CR.xlsx",   
 sheet = "MIP 2017",   
 rows= c(12:264),   
 cols = c(4:256),   
 skipEmptyRows = FALSE,   
 colNames = FALSE,   
 rowNames = FALSE)  
 ) # <-- Fin del paréntesis  
  
nombres <- read.xlsx("MIP-AE-AE-017-CR.xlsx",   
 sheet = "MIP 2017",   
 rows= c(12:264),   
 cols = 1,   
 skipEmptyRows = FALSE,   
 colNames = FALSE,   
 rowNames = FALSE  
) # <-- Fin del paréntesis  
  
  
# Agregamos nuestra matriz para tener un valor por actividad  
  
# Nombramos nuestra matriz Z  
colnames(Z\_cruda) <- as.vector(nombres$X1)  
rownames(Z\_cruda) <- as.vector(nombres$X1)  
  
Z <- aggregate.Matrix(Z\_cruda, as.factor(nombres$X1),fun = "sum")  
Z <- t(aggregate.Matrix(t(Z), as.factor(nombres$X1),fun = "sum"))  
Z <- as.matrix(Z)  
  
dim(Z)

## [1] 136 136

# Demanda Final  
DF\_cruda <- as.matrix(read.xlsx("MIP-AE-AE-017-CR.xlsx",   
 sheet = "MIP 2017",   
 rows= c(12:264),   
 cols = c(258:262),   
 skipEmptyRows = FALSE,   
 colNames = FALSE,   
 rowNames = FALSE)  
 ) # <-- Fin del paréntesis  
  
  
# Lo mismo para la demanda final  
codsDF <- as.matrix(  
 t(  
 read.xlsx(  
 "MIP-AE-AE-017-CR.xlsx",  
 sheet = "MIP 2017",   
 rows= c(9:10),   
 cols = c(258:262),   
 skipEmptyRows = FALSE,  
 colNames = FALSE,   
 rowNames = FALSE  
 ))  
)# <-- Fin del paréntesis  
  
# Y nuestra matriz de demanda final DF  
colnames(DF\_cruda) <- as.vector(codsDF[,1])  
rownames(DF\_cruda) <- as.vector(rownames(Z\_cruda))  
  
# Y agregamos las filas que se repiten  
DF <- aggregate.Matrix(DF\_cruda, as.factor(rownames(DF\_cruda)),fun = "sum")  
DF <- as.matrix(DF)  
  
dim(DF)

## [1] 136 5

# =============================================================================  
# Cuenta de energía  
  
# Importamos los datos crudos  
E\_cruda <- as.matrix(read.xlsx("COUF-2017.xlsx",   
 sheet = "COUF-E 2017",   
 rows= c(127:146),   
 cols = c(3:169),   
 skipEmptyRows = FALSE,   
 colNames = FALSE,   
 rowNames = TRUE # Sí hay nombres de fila  
 )  
) # <-- Fin del paréntesis  
  
# Extraemos los nombres de columna  
nombres\_e <- t(read.xlsx("COUF-2017.xlsx",   
 sheet = "COUF-E 2017",   
 rows= c(16),   
 cols = c(4:169),   
 skipEmptyRows = FALSE,   
 colNames = FALSE,   
 rowNames = FALSE)  
 ) # <-- Fin del paréntesis  
  
# Y nombramos las columnas de nuestra matriz de usos energéticos  
colnames(E\_cruda) <- c(nombres\_e[,1])  
  
# Las dimensiones de E\_cruda son mayores a las de Z  
# porque hay agregaciones por grupos de sectores y  
# hay sectores desagregados a mayor detalle.  
  
dim(E\_cruda)

## [1] 20 166

# Identificamos las posiciones que son sumas de sectores  
# Nótese que dejamos dos sumas dentro que no tienen detalle  
# correspondientes a AE082 (Electricidad) y AE144 (hogares como empl.)  
  
posGruposEnergia <- c(1,31,35,78,82,94,97,106,110,113,  
 119,122,133,143,147,150,153,158)  
  
# Extraemos solo los sectores (nótese el "-" antes de posGruposEnergia)  
E\_cruda <- E\_cruda[ , -posGruposEnergia]  
  
# Utilizando la función substr() extraemos los primeros 5 digitos de  
# la nomenclatura para poder agregar por actividades que comparten  
# esos mismos.  
colnames(E\_cruda) <- substr(colnames(E\_cruda), start = 1, stop = 5)  
  
# Y agregamos utilizando el mismo procedimiento que anteriormente.  
E <- as.matrix(t(aggregate.Matrix(t(E\_cruda), colnames(E\_cruda),fun = "sum")))  
E <- as.matrix(E)  
# Y chequeamos que nuestras dimensiones sean iguales a las columnas  
# de Z  
dim(E)

## [1] 20 136

# Para ser congruentes con Z, renombramos las columnas con los nombres  
# completos de Z  
colnames(E) <- colnames(Z)  
  
# Hacemos limpieza  
rm(Z\_cruda,DF\_cruda,E\_cruda, nombres\_e)  
  
# =============================================================================  
## Cuenta de energía 2018 (Preliminar No-Citar)  
  
# Importamos los datos crudos  
E\_cruda <- as.matrix(read.xlsx("COUF-2018.xlsx",   
 sheet = "COUF-E 2018",   
 rows= c(126:145),   
 cols = c(3:170),   
 skipEmptyRows = FALSE,   
 colNames = FALSE,   
 rowNames = TRUE # Sí hay nombres de fila  
 )  
) # <-- Fin del paréntesis  
  
# Extraemos los nombres de columna  
nombres\_e <- t(read.xlsx("COUF-2018.xlsx",   
 sheet = "COUF-E 2018",   
 rows= c(15),   
 cols = c(4:170),   
 skipEmptyRows = FALSE,   
 colNames = FALSE,   
 rowNames = FALSE)  
) # <-- Fin del paréntesis  
  
# Y nombramos las columnas de nuestra matriz de usos energéticos  
colnames(E\_cruda) <- c(nombres\_e[,1])  
  
# Las dimensiones de E\_cruda son mayores a las de Z  
# porque hay agregaciones por grupos de sectores y  
# hay sectores desagregados a mayor detalle.  
  
dim(E\_cruda)

## [1] 20 167

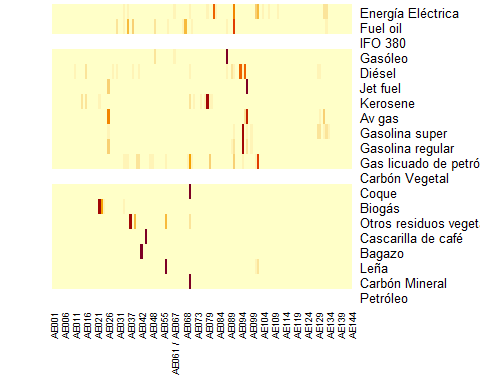
# Identificamos las posiciones que son sumas de sectores  
# Nótese que dejamos dos sumas dentro que no tienen detalle  
# correspondientes a AE082 (Electricidad) y AE144 (hogares como empl.)  
  
# Notar que respecto de 2017 lo siguiente cambia a partir de 106 (107)  
  
posGruposEnergia <-c(1,31,35,78,82,94,97,107,111,114,  
 120,123,134,144,148,151,154,159)  
  
# Extraemos solo los sectores (nótese el "-" antes de posGruposEnergia)  
E\_cruda <- E\_cruda[ , -posGruposEnergia]  
  
# Utilizando la función substr() extraemos los primeros 5 digitos de  
# la nomenclatura para poder agregar por actividades que comparten  
# esos mismos.  
colnames(E\_cruda) <- substr(colnames(E\_cruda), start = 1, stop = 5)  
  
# Y agregamos utilizando el mismo procedimiento que anteriormente.  
E18 <- as.matrix(t(aggregate.Matrix(t(E\_cruda), colnames(E\_cruda),fun = "sum")))  
E18 <- as.matrix(E18)  
# Y chequeamos que nuestras dimensiones sean iguales a las columnas  
# de Z  
dim(E18)

## [1] 20 136

# Para ser congruentes con Z, renombramos las columnas con los nombres  
# completos de Z  
colnames(E18) <- colnames(Z)  
  
# Hacemos limpieza  
rm(E\_cruda)  
  
moltenE18 <- as.matrix(melt(E18))  
  
# Exportamos a Excel  
write.xlsx( as.data.frame(moltenE18) ,   
 "CuentaEnergiaBD\_2018.xlsx",  
 sheetName= "datos",  
 startRow = 5,  
 startCol = 1,  
 asTable = FALSE,   
 colNames = TRUE,   
 rowNames = TRUE,   
 overwrite = TRUE  
)  
  
# =============================================================================  
# Modelo de insumo producto  
  
# Producción  
x <- as.vector(rowSums(Z) + rowSums(DF))  
  
# Demanda final  
f <- as.vector(rowSums(DF))  
  
# x sombrero  
xhat <- diag(x)  
xhat\_inv <- solve(xhat)  
  
# Matriz de coeficientes técnicos  
A <- Z %\*% solve( xhat )  
  
# Matriz identidad  
I <- diag( dim(A)[1])  
  
# Matriz de Leontief  
L <- solve(I - A )  
  
# Coeficientes de uso de cada energético por unidad de producto  
EC <- E %\*% solve(xhat)  
colnames(EC) <- colnames(Z)  
  
# Nueva demanda final  
f1 <- f  
f1[80] <- f1[80] \*1.20  
  
# Cálculo de nuevas demandas de energía por energético  
EC %\*% L %\*% f1

## [,1]  
## Petróleo 0.000000  
## Carbón Mineral 217.854446  
## Leña 6155.246485  
## Bagazo 9361.769370  
## Cascarilla de café 400.455355  
## Otros residuos vegetales 3336.413576  
## Biogás 4.442436  
## Coque 3855.035343  
## Carbón Vegetal 0.000000  
## Gas licuado de petróleo 5048.034741  
## Gasolina regular 5510.603805  
## Gasolina super 4471.370715  
## Av gas 52.332118  
## Kerosene 232.172537  
## Jet fuel 1112.688822  
## Diésel 38229.913051  
## Gasóleo 426.222919  
## IFO 380 0.000000  
## Fuel oil 5059.333812  
## Energía Eléctrica 22498.552276

# Diferencias  
deltaE <- cbind( rowSums(E),   
 (EC %\*% L %\*% f1),   
 (EC %\*% L %\*% f1)- rowSums(E),   
 ((EC %\*% L %\*% f1)- rowSums(E))\*100/ rowSums(E)   
 ) # <-- fin del paréntesis  
colnames(deltaE) <- c("Original", "Política", "Diferencia", "Porcentual")  
  
# Y si queremos el detalle  
E1 <- EC %\*% diag(as.vector(L %\*% f1))  
colnames(E1) <- colnames(E)  
  
# =============================================================================  
# Bioeconomía  
  
# Clasificación cruzada bioeconomía AECR  
clasifs\_bioeconomia <- read.xlsx("clasificacion\_cruzada\_bioeconomia.xlsx",   
 sheet = "datos",   
 rows= c(1:137),   
 cols = c(1:6),   
 skipEmptyRows = FALSE,   
 colNames =TRUE,   
 rowNames = FALSE  
) # <-- Fin del paréntesis  
  
  
  
  
  
# =============================================================================  
# Excel  
  
write.xlsx(as.data.frame(colnames(Z)),  
 "nombres\_Z.xlsx",  
 sheetName = "datos",  
 startRow = 5,  
 startCol = 1,  
 asTable = FALSE,  
 colNames = TRUE,  
 rowNames = TRUE,  
 overwrite = TRUE)  
  
# Cambios en datos ambientales  
write.xlsx( as.data.frame(deltaE) ,   
 "datos\_ambientales.xlsx",  
 sheetName= "datos",  
 startRow = 5,  
 startCol = 1,  
 asTable = FALSE,   
 colNames = TRUE,   
 rowNames = TRUE,   
 overwrite = TRUE  
 )  
  
# Gráfico  
heatmap(E1, Colv = NA, Rowv = NA)



# # =============================================================================  
# # Bloopers  
#   
# # Inicialmente creímos que la matriz Z tenía un componente importado  
# # y un componente nacional. Ese no es el caso, solo está dividida por  
# # el tipo de control "nacional" o "extranjero" de la producción nacional.  
# # Gracias a Johnny Aguilar por la aclaración.  
#   
# # Dejo la solución para extraer filas y columnas con índices, solamente  
# # porque es una buena ilustración de algo que se utiliza continuamente  
# # en el trabajo con este tipo de datos.  
#   
# # Obtenemos nuestros códigos de actividad y nombres  
# cods <- as.data.frame(  
# read.xlsx(  
# "MIP-AE-AE-017-CR.xlsx",  
# sheet = "clasificaciones",   
# rows= c(5:141),   
# cols = c(1:5),   
# skipEmptyRows = FALSE,   
# colNames = TRUE,   
# rowNames = FALSE  
# )  
# )# <-- Fin del paréntesis  
#   
# # componente doméstico  
# local <- c(2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,   
# 32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,   
# 62,64,66,68,70,71,73,75,77,78,80,81,83,85,87,88,   
# 90,92,94,95,97,98,99,100,101,103,105,107,108,110,   
# 112,114,115,117,119,120,121,123,125,127,129,131,   
# 133,135,137,139,141,143,145,147,149,151,153,155,   
# 157,159,161,162,164,166,167,169,171,173,174,176,   
# 178,180,182,184,186,188,190,192,194,196,197,199,   
# 200,202,204,206,208,210,212,214,216,218,219,221,   
# 223,225,227,229,231,233,235,237,239,241,243,245,   
# 247,249,251,253)  
#   
# # Si el BCR hubiera dejado (aunque en ceros) el componente importado  
# # de los sectores que solamente tienen componente local, los indices  
# # serían más fáciles de construir con:  
# # seq( from=2, to= 254, by=2 ) para lo doméstico   
#   
# # Aquí utilizamos el método del índice en R para obtener nuestros elementos  
#   
# # Compras entre actividades económicas domésticas (consumo intermedio)  
# Z <- Z\_cruda[ local , local ]  
#   
# # Compras de producto importado de las actividades domésticas (importaciones)  
# M <- Z\_cruda[-local,local]  
#   
# # Compras de Actividades económicas no-domésticas a las actividades locales  
# # (exportaciones)  
# X <- Z\_cruda[local,-local]  
#   
# # Compras de Actividades no-domésticas a no-domésticas?  
# XX <- Z\_cruda[-local,-local]